

**Amendments to the Specification:**

Please replace the paragraph beginning at page 1, line 20 to line 28, with the following rewritten paragraph:

Multi-vendor security services and products typically provide network security. Each security service and product typically must be configured and reconfigured to maintain network security. Typically, each vendor for a security service and product utilizes various settings to establish a configuration. Some of these ~~setting~~ settings may be established at the time a product is installed. A user, a group, or update features of the product may establish other settings.

Please replace the paragraph beginning at page 2, line 10 to line 22, with the following rewritten paragraph:

This problem is exacerbated because there is not a common technique or structure for maintaining configuration data for the wide variety of products, devices and groupings of products and devices found in the complex enterprise networks that exist today. For this reason, an integrated approach to configuring devices and products to support network security management is a very complex and difficult to implement. Moreover, an approach developed for one enterprise network may not be applicable to another enterprise network due to the differences in the network structure, the security products and services, and the use of the network.

Please replace the paragraph beginning at page 7, line 2 to line 6, with the following rewritten paragraph:

Fig. 1 is an illustration of a system that utilizes a configuration system and methods including configuration

inheritance and revisioning for managed products according to one embodiment of the present invention.

Please replace the paragraph beginning at page 9, line 13 to line 16, with the following rewritten paragraph:

Fig. 12 is an example of an XML string that is generated in one embodiment of the get trunk object and setting objects operation of Fig. 11 according to the principles of this invention.

Please replace the paragraph beginning at page 9, line 17 to line 23, with the following rewritten paragraph:

Figs. 13A and 13B are a graphical representation of a DOM tree corresponding to the XML string of Fig. 12, a derived DOM tree, with the collision detection names corresponding to start tags in the XML strings presented in the nodes associated with elements having the start tag according to one embodiment of the present invention.

Please replace the paragraph beginning at page 9, line 24 to line 27, with the following rewritten paragraph:

Fig. 14 is an example of an XML string that is generated in one embodiment of the get parent object and setting objects operation of Fig. 11 according to the principles of this invention.

Please replace the paragraph beginning at page 9, line 28 to line 34, with the following rewritten paragraph:

Figs. 15A and 15B are a graphical representation of a DOM tree corresponding to the XML string of Fig. 14, a base DOM

tree with the collision detection names corresponding to start tags in the XML strings presented in the nodes associated with elements having the start tag according to one embodiment of the present invention.

Please replace the paragraph beginning at page 10, line 5 to line 7, with the following rewritten paragraph:

Figs. 17A and 17B are an XML string that is generated from the DOM tree of Figs. 16A and 16B according to one embodiment of the present invention.

Please replace the paragraph beginning at page 11, line 15 to line 22, with the following rewritten paragraph:

As explained more completely below, a parent-child inheritance merge process resolves collisions between elements having the same collision detection name for each pair of parent/child configuration objects and merges the pair of configuration objects to obtain a merged configuration. When the last configuration object in a family has been processed, the resulting merged configuration is the effective configuration.

Please replace the paragraph beginning at page 13, line 21 to line 28, with the following rewritten paragraph:

Thus, the event configuration object is converted to a data string by selecting the correct revision of each of its setting objects and concatenating the data string from each setting. The resulting configuration string is merged with its parent configuration, and so on with the grandparent, great-grandparent, etc. using a parent-child inheritance merge process, as described below.

Please replace the paragraph beginning at page 14, line 25 to line 32, with the following rewritten paragraph:

Management server 140 receives, interprets and serializes all the service requests made via events from managed products 110\_1 to 110\_nk and ~~route~~ routes the events to the appropriate event sink or sinks, e.g., a database server, whether it is a directory server 115A, log database server 110A, an alert event database server 110B, and/or security and feedback control system 155.

Please replace the paragraph beginning at page 17, line 19 to line 32, with the following rewritten paragraph:

As explained in copending, commonly filed, and commonly assigned U.S. Patent Application Serial No. 10/~~xxx,xxx~~ 10/660,225, (Attorney Docket No. SYMC1001) entitled "A SECURITY MANAGEMENT SYSTEM INCLUDING FEEDBACK AND CONTROL" of Paul M. Agbabian, which is incorporated herein by reference in its entirety, a managed product is registered with management server 140 by utilizing a product integration XML (PIX) file. Product integration data included in the PIX files includes a product name, a product identifier (ID), and a software feature identifier as well as settings and properties associated with the software feature(s) of the managed product. A software feature represents a particular function or capability of a managed product.

Please replace the paragraph beginning at page 42, line 12 to line 25, with the following rewritten paragraph:

Copending, commonly filed, and commonly assigned U.S. Patent Application Serial No. 10/~~xxx,xxx~~ 10/660,225, (Attorney

Docket No. SYMC1001) entitled "A SECURITY MANAGEMENT SYSTEM INCLUDING FEEDBACK AND CONTROL" of Paul M. Agbabian provides a description of the elements of both Fig. 1 and Fig. 8 as well as event structures, files, etc. that are utilized by system 100. The Summary, Brief Description of the Drawings, Detailed Description, Claims and Abstract and Drawings of U.S. Patent Application Serial No. 10/~~xxx,xxx~~ 10/660,225, (Attorney Docket No. SYMC1001) entitled "A SECURITY MANAGEMENT SYSTEM INCLUDING FEEDBACK AND CONTROL" of Paul M. Agbabian are incorporated herein by reference in their entireties.

Please replace the paragraph beginning at page 49, line 9 to line 18, with the following rewritten paragraph:

Parent check operation 1103 determines whether there is a parent configuration object, for the derived configuration object, e.g., does the parent configuration pointer field **DN\_parent** of the derived object include a valid pointer. In the example of Fig. 10, configuration object 900 has a valid point to parent configuration object 600 in parent configuration pointer field **DN\_parent**. Thus, parent check operation 1103 transfers to get parent object and setting objects operation 1104.

Please replace the paragraph beginning at page 61, line 32 through to page 62, line 8, with the following rewritten paragraph, please note an underscore line is present in the corrected identifier (i.e., 1350\_1) that is barely evident due to the underlining of the added material:

Source start tag 1240\_2 has two child start tags 1250\_2 and 1250\_3 with corresponding child nodes 1350\_2 and 1350\_3 of node 1340\_2. Nodes 1350\_2 and 1350\_3 have the collision detection names of IP1.0.0.0 and IP3.0.0.0, respectively. This

is because the name attribute is used in both IP start tag 1250\_2 and 1250\_3 with a current element text **namespecifier** of "=" that appends the current element text to the IP start tag name. (See TABLE 15.) All other nodes 1340\_1, 1340\_3 to 1340\_5, ~~1350\_1~~ 1350\_1, 1350\_4 to 1350\_6, and 1360\_1 to 1360\_3 related to node 1330\_1 have default names based upon the corresponding start tag or text element.

Please replace the paragraph beginning at page 64, line 13 to line 24, with the following rewritten paragraph, please note an underscore line is present in the corrected identifier (i.e., 1550\_4) that is barely evident due to the underlining of the added material:

Source start tag 1440\_2 has two child start tags 1450\_2 and 1450\_3 with corresponding child nodes 1550\_2 and 1550\_3 of node 1540\_2. Nodes 1550\_2 and 1550\_3 have the collision detection names of IP2.0.0.0 and IP1.0.0.0, respectively. This is because the name attribute is used in both IP start tag 1450\_2 and 1450\_3 with a current element text **namespecifier** of "=" that appends the text element of the tag to the IP start tag name. All other nodes 1540\_1, 1540\_3 to 1540\_7, 1550\_1, ~~1550\_4~~ 1550\_4 to 1550\_8, and 1560\_1 to 1560\_5 have default names based upon the corresponding start tag or text.